This Page Is Inserted by IFW Operations and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents will not correct images, please do not report the images to the Image Problem Mailbox.

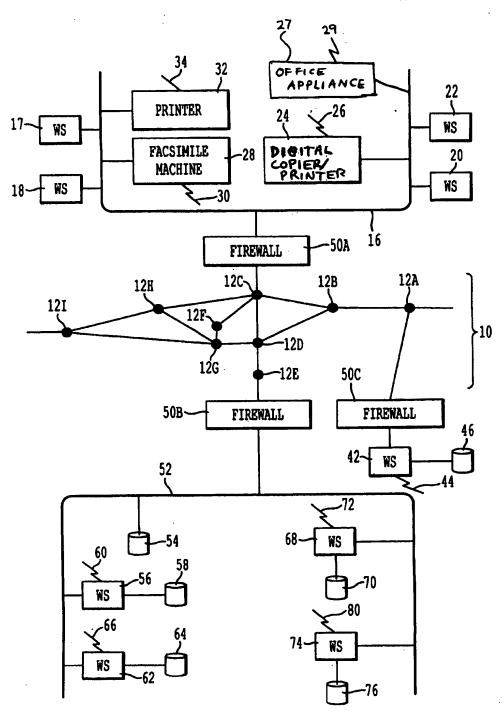


FIG. 1

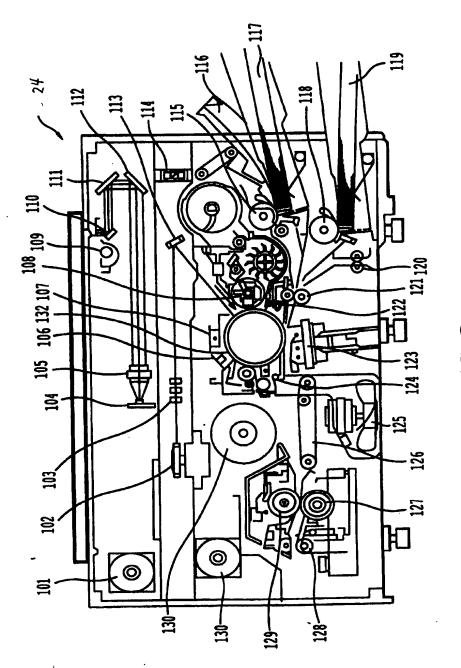


FIG. 2

Inventor: Tetsuro MOTOYAMA, et al.

SHEET 3 of 39

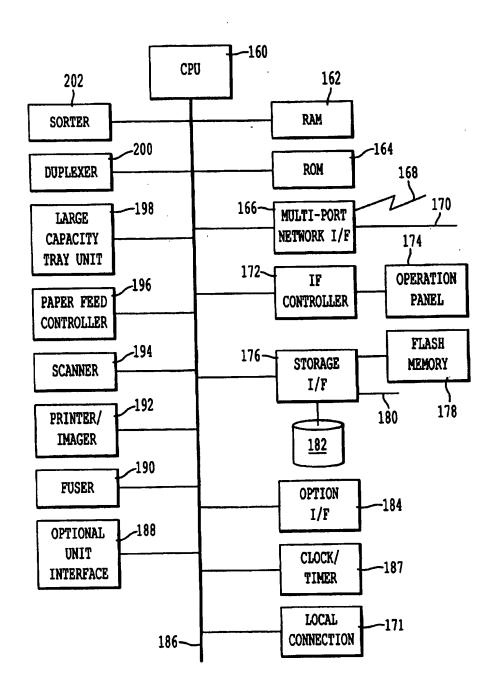


FIG. 3

OBLON, SPIVAK, et al.

Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 4 of 39

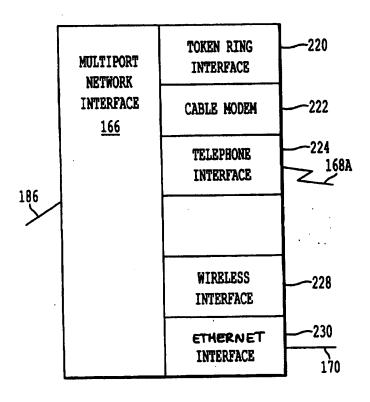


FIG. 4

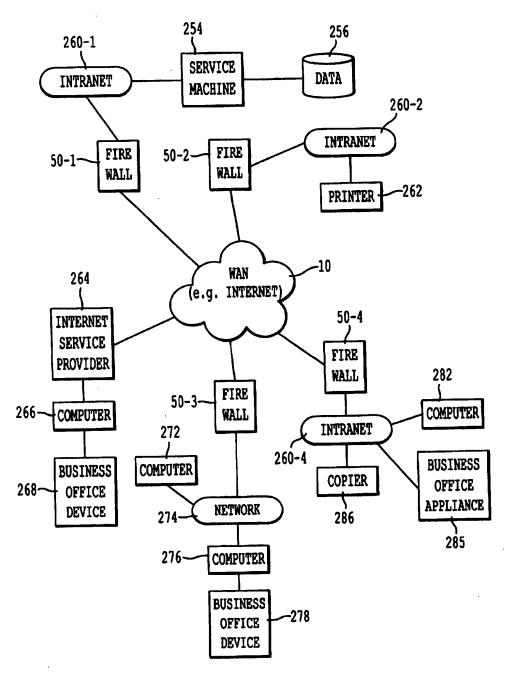


FIG. 5

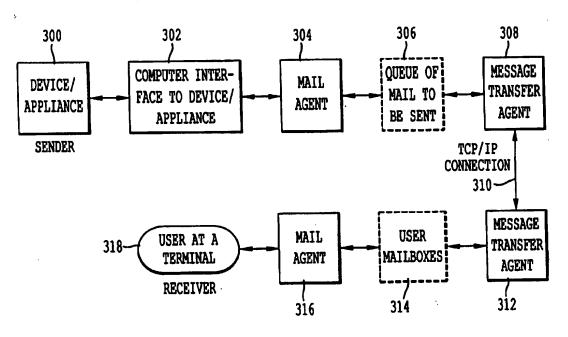


FIG. 6A

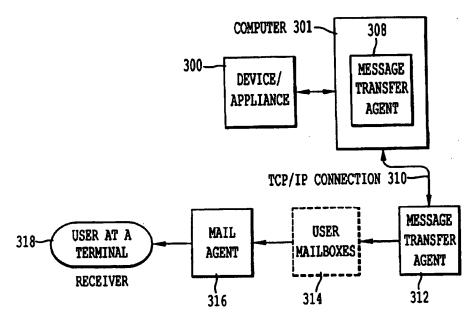


FIG. 6B

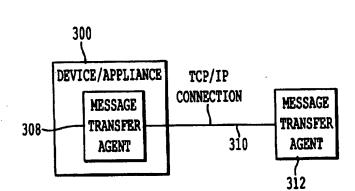


FIG. 6C

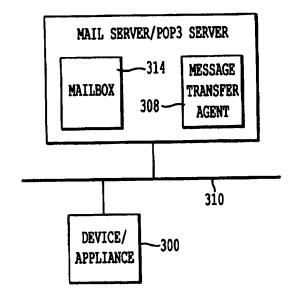
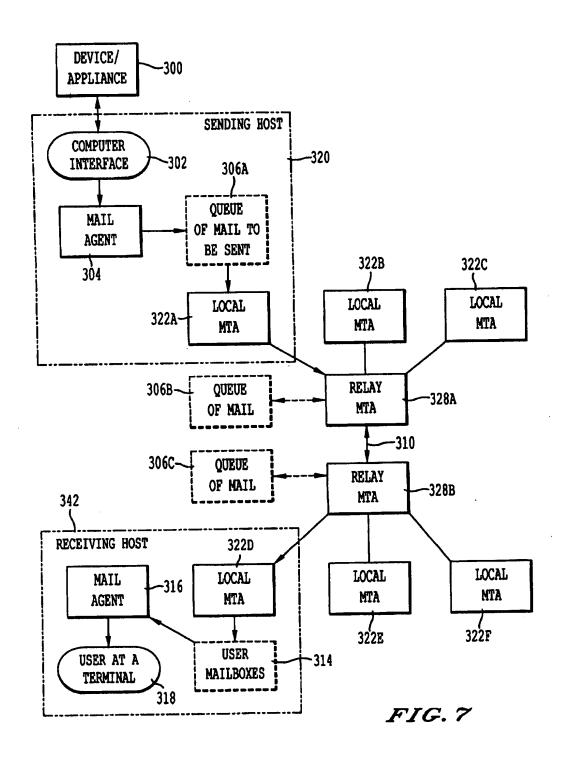


FIG. 6D



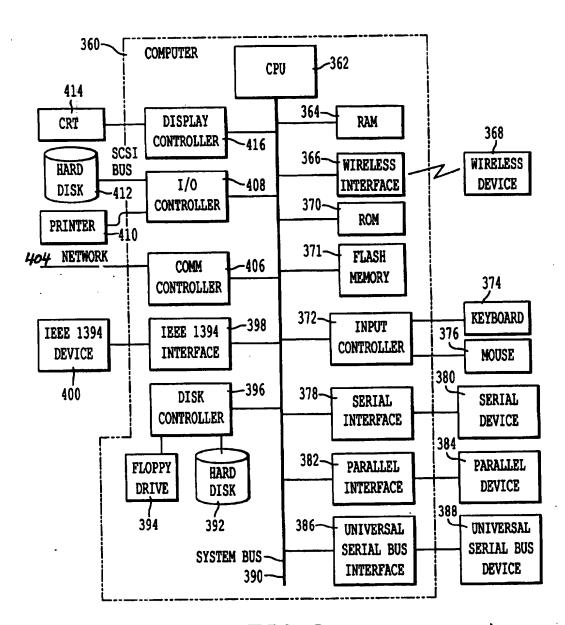


FIG. 8

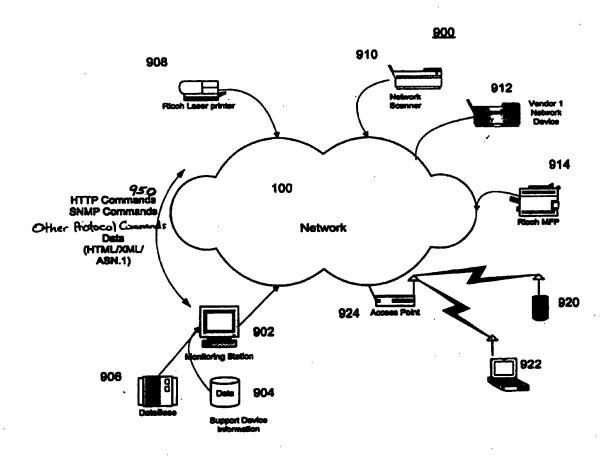


Figure 9

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 11 of 39

1000

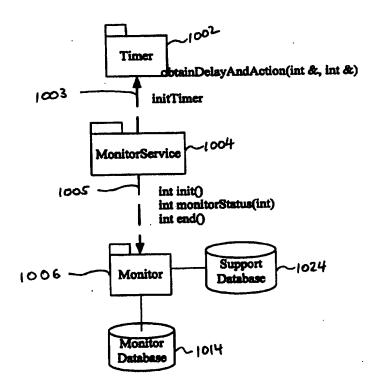


FIG. 10

Use typedef to define infoType to be int type updateConfig(map<infoType, string> &) -1101 Common bool obtainConfig(map<infoType, string> &, map<string, vector<SParameter>>&) bool saveStatus(map<infoType, string> &) DeviceODBC MonitorManage: 1102 CDevice * createDevice(const string & in_sIP, CHWaccess & in_HWaccess, map<string, vector<SParameter>> & in ProtocolParameters) bool canAccessHW(void) bool getVendor(string & out_sVendor) bool getModel(string & out_sModel) bool getUniqueID(string & out_sUniqueID) bool obtainStatus(map<infoType, string> & out_StatusMap) enum checkEventStatus(void) bool obtain Event Status 1116 map<infoType, string> & out_EventStatusMap) void clearEventStatus(void) Device HWaccess. -1110 enum return for checkEventStatus void initBegin(void) should have values eNoEventSinceClearAndNoEventDetected, void initEnd(void) eNoEventSinceClearAndEventDetected, bool canAccessIP(const string & in_sIP, eEventSinceClearAndNoEventDetected, map<string, vector<SParameter>> & in_ProtocolParameters) eEventSinceClearAndEventDetected. bool obtainVendor(string & out_sVendor, map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sIP) bool obtainModel(string & out_sModelName, map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sIP) bool obtainUniqueID(string & out_sUniqueID, map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sIP) EerrorCode obtainEventStatus(map<infoType,string> & out_StatusMap, const string & in_sIP, map<string, vector<SParameter> > & in_ProtocolParameters) bool obtainStatus(map<infoType,string> & out_StatusMap, const string & in_sIP, const string & in_sVendor, const string & in_sModel, map<string, vector<SParameter> > & in_ProtocolParameters)

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 13 of 39

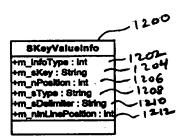


FIG. 12

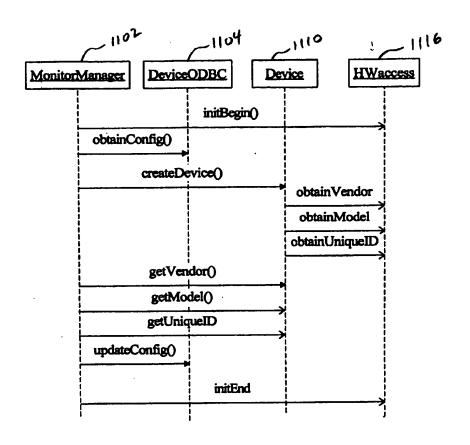


FIG. 13

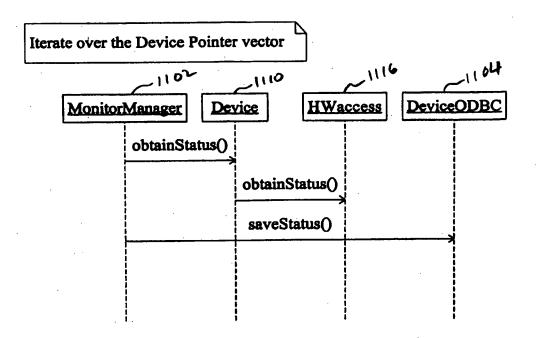


FIG. 14

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 16 of 39

1500

1502	۱۶۵ منر	4		
Pointer to	Pointer to	Pointer to	Pointer to	• • • • • • • • •
CDevice	CDevice	CDevice	CDevice	
Object	Object	Object	Object	

std::vector <pointer of CDevice >

FIG. 15

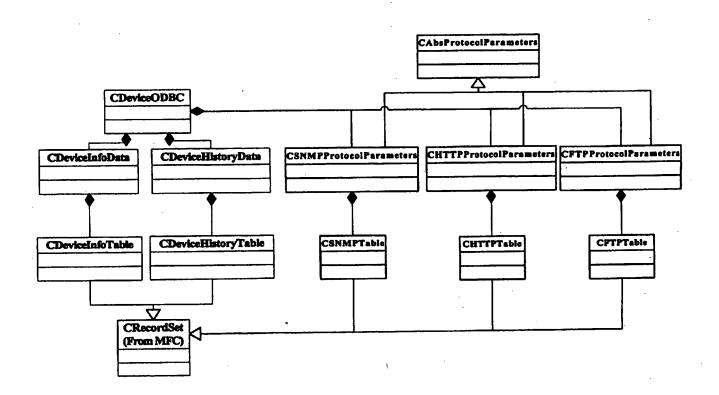


FIG. 16

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 18 of 39

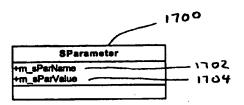


FIG. 17

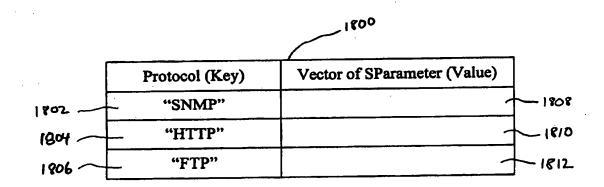
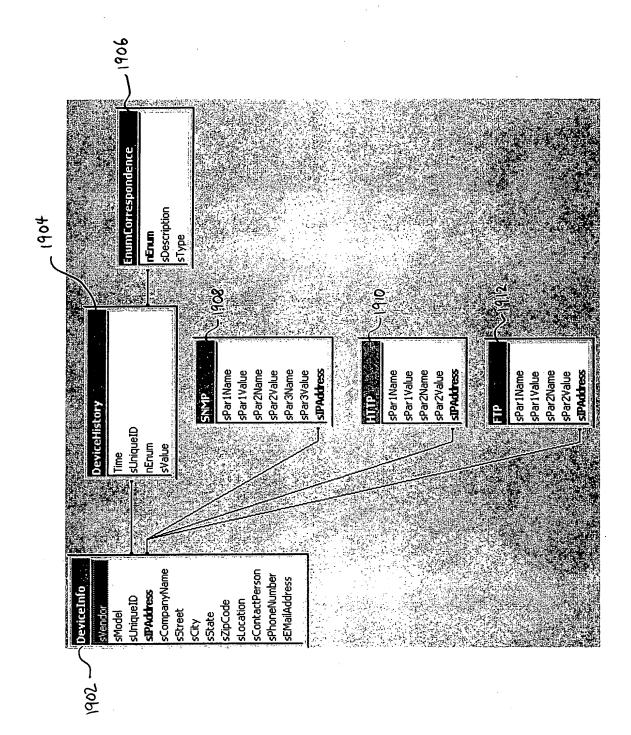


FIG. 18



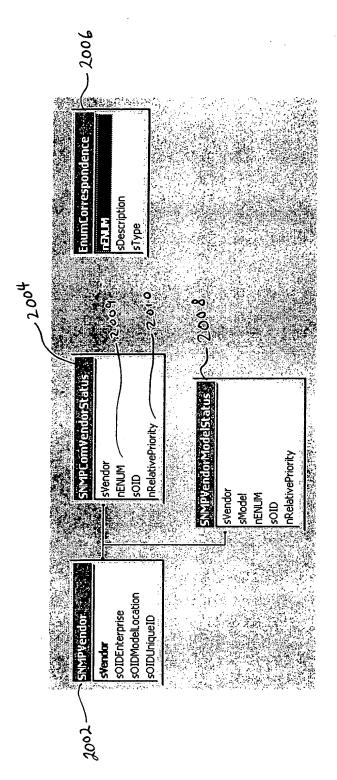


FIG. 20

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 21 of 39

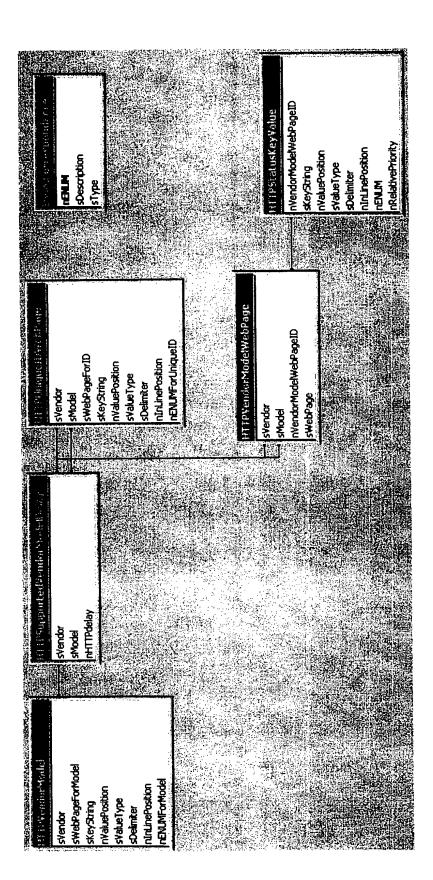


FIG. 2

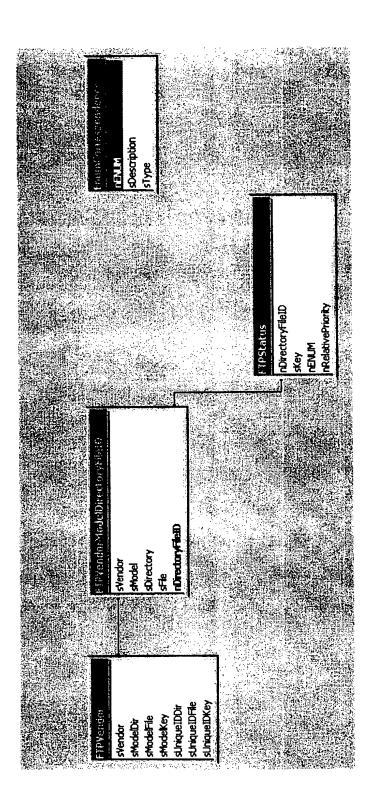


FIG. 27

2300 CHWaccess void initBegin(void) void initEnd(void) bool canAccessIP(const string & in_sIP, map<string, vector<SParameter> > & in_ProtocolParameters) bool obtainVendor(string & out_sVendor, map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sIP) bool obtainModel(string & out_sModelName, map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sIP) bool obtainUniqueID(string & out_sUniqueID, map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sIP) EerrorCode obtainEventStatus(map<infoType, pair<std::string, int> > & inOut_StatusMap, const string & in_sIP, map<string, vector<SParameter> > & in_ProtocolParameters) bool obtainStatus(map<infoType, pair<string, int> > & inOut_StatusMap, const string & in_sIP, const string & in_sVendor, const string & in_sModel, map<string, vector<SParameter> > & in_ProtocolParameters) void initWithVendor(map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sVendor) void initWithModel(map<string, vector<SParameter> > & inOut_ProtocolParameters, const string & in_sModel) 2302 2304 2306 HTTP SNMP FTP 2308 2310 CRecordSet(From MFC) CAbsProtocol

FIG. 23

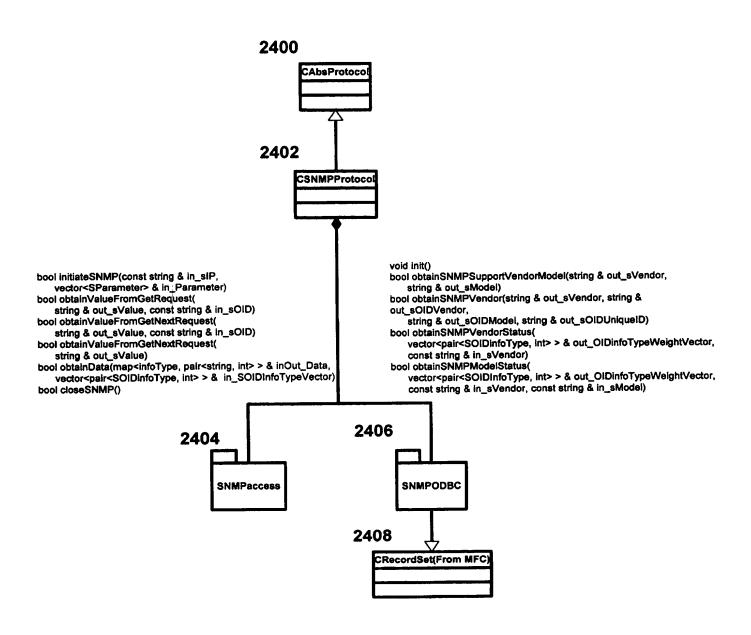


FIG. 24

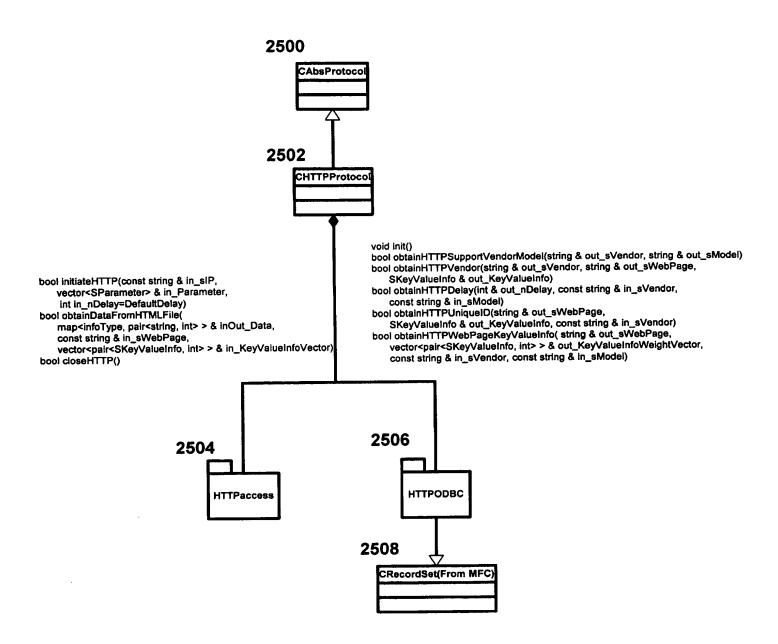


FIG. 25

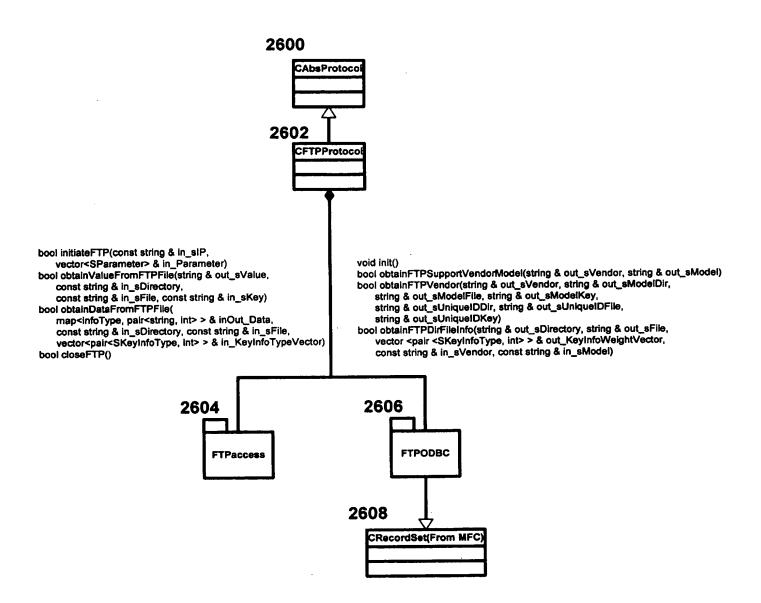


FIG. 26

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 27 of 39

Vector of CAbsProtocol* 500

f T			
CAbsProtocol* 502	CAbsProtocol* 504	CAbsProtocol* 506	•••

FIG. 27A

std::map<std::string, std::wector<std::pair<SOIDinfoType, int>>>> 510

-4	İ	map 514		
string 512	string 516	vector 518		
	GENERIC	<(SOIDinfoType1, int1), (SOIDinfoType2, int2) >		
Ricoh	Aficio 1224C	<(SOIDinfoType3, int3)>		
	GENERIC	<(SOIDinfoType4, int4), (SOIDinfoType5, int5) >		
Xerox	DocuPrint NC60	<(SOIDinfoType5, int5)>		

FIG. 27B

std::map<std::string, std::map<std::string, std::vector<SWebPageInfo>>> 520

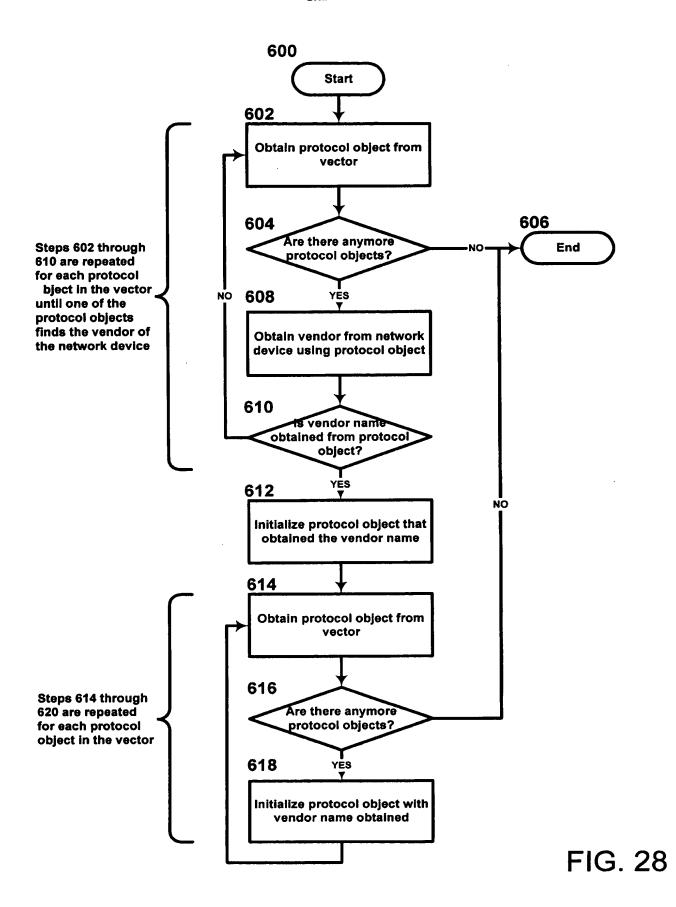
string		map 524	
522	string 526	vector 528	
Ricoh	Aficio X	struct SWebPageInfo { std::string m_sWebPage; vector of std::vector <std::pair<skeyvalueinfo, int="">> m_KeyValueInfoVector; };</std::pair<skeyvalueinfo,>	
Xerox	N4025		

FIG. 27C

std::map<std::string, std::map<std::string, std::vector<SDirFileStatusInfo>>> 530

string		map 534
532	string 536	vector 538
Ricoh	Aficio	struct SDirFileStatusInfo {
Xerox	N4025	

FIG. 27D



OBLON, SPIVAK, et al. Docket No: 245418US-2

Inventor: Tetsuro MOTOYAMA, et al.

706
struct SOIDinfoType {
infoType m_infoType;
std::string m_sOID;
SOIDInfoType();
~SOIDinfoType();
void clear();

714

struct SKeyValueInfo {
 infoType m_infoType;
 std::string m_sKey;
 int m_nPosition;
 std::string m_sType;
 std::string m_sDellmiter;
 int m_nInLinePosition;

SKeyValueInfo(); ~SKeyValueInfo(); void clear();

722

struct SKeyInfoType {
 infoType m_InfoType;
 std::string m_sKey;
 SKeyInfoType();
 ~SKeyInfoType();

void clear();

};

SHEET 30 of 39

	SNMP Protocol 700	
pair <soidinfotype, int=""> 702</soidinfotype,>	pair <soidinfotype, int=""></soidinfotype,>	
	FIG. 29A	

HTTP Protocol 708

710 712	pair <skeyvalueinfo, int=""></skeyvalueinfo,>	pair <skeyvalueinfo, int=""></skeyvalueinfo,>	• • •
-----------	---	---	-------

FIG. 29B

FTP Protocol 716

pair <skeyinfotype, int=""> pair<skeyinfotype, int=""></skeyinfotype,></skeyinfotype,>	pair <skeyinfotype, int=""></skeyinfotype,>	SKeyInfoType, int>
--	---	--------------------

FIG. 29C

Status Information Map 724

726	728
infoType	pair <string, int=""></string,>
infoType	pair <string, int=""></string,>
infoType	pair <string, int=""></string,>

FIG. 29D

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 31 of 39

std::map<std::string, std::wector<SDirFileStatusInfo>>>

	Aficio 120	<sdirfilestatusinfo1, sdirfilestatusinfo2,="" sdirfilestatusinfo3=""></sdirfilestatusinfo1,>
Ricoh		
		802
/pub		
status.txt	· · · · · · · · · · · · · · · · · · ·	
<(SKeyInfoT)	ype1, 1000), (S pe5, 625)>	KeylnfoType2, 5000), (SKeylnfoType3, 10000), (SKeylnfoType4, 7500),

SKeyInfoType1 corresponds to the infoType 600, SKeyInfoType2 corresponds to the infoType 610, SKeyInfoType3 corresponds to the infoType 620, SKeyInfoType4 corresponds to the infoType 700, and SKeyInfoType5 corresponds to the infoType 710

Status Information Map

804

600	("Low Paper", 500)
610	("24321", 10000)
700	("OK", 2500)

FTP Protocol 806

(SKeyinfoType1, 1000)	(SKeyInfoType3, 10000)	(SKeyInfoType4, 7500)	(SKeyinfoType5, 625)

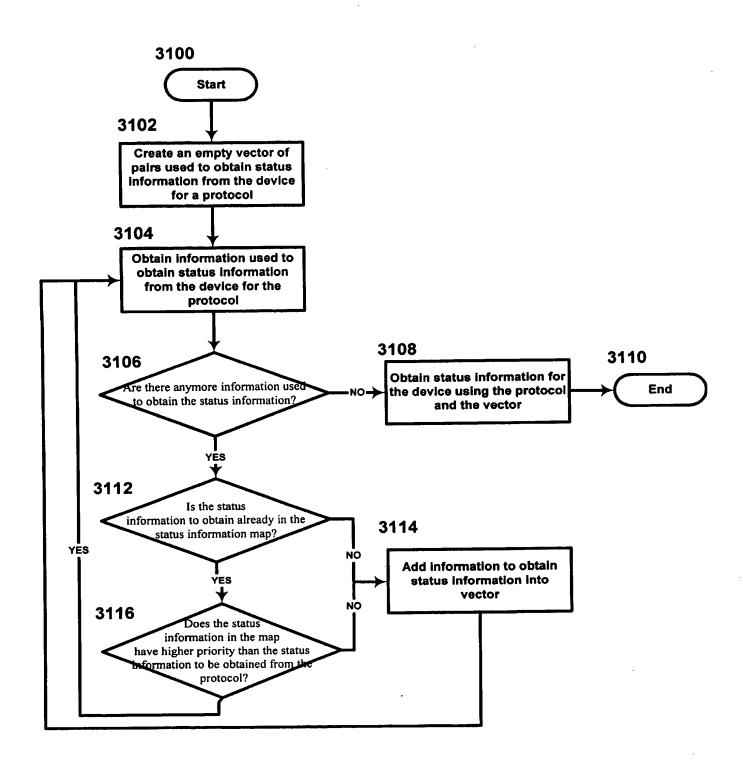


FIG. 31A

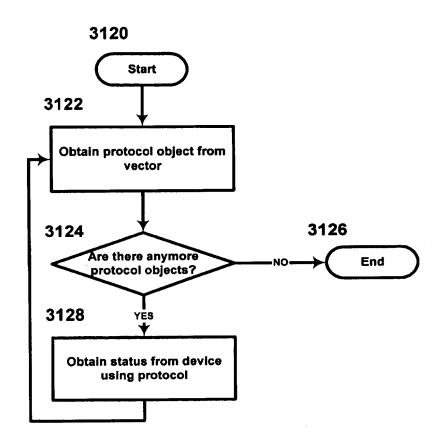


FIG. 31B

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 34 of 39

Vendor Model Support Map 3200

3202 3204

string	int
string	int
string	int
string	int
	• • •

FIG. 32A

Sample Vendor Model Support Map 3206

3208 3210
Xerox%%%%NC60 1

X010X /6 /6 /6 /6 /6 INCOU		
Xerox%%%%%N4025	1	
HP%%%%%LaserJet 9000	1	
HP%%%%%LaserJet 4550	· 1	

FIG. 32B

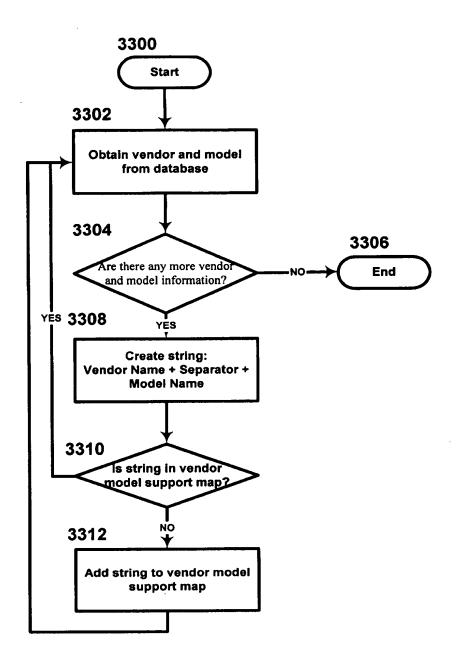


FIG. 33

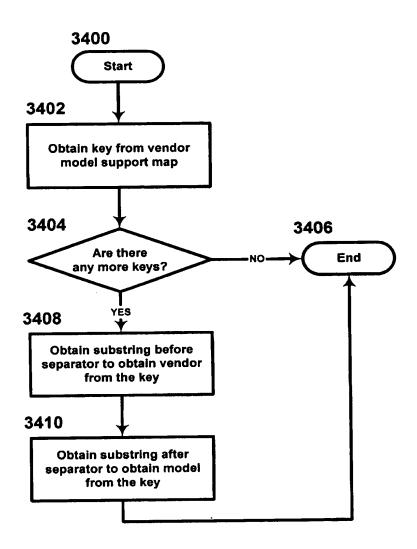


FIG. 34

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 37 of 39

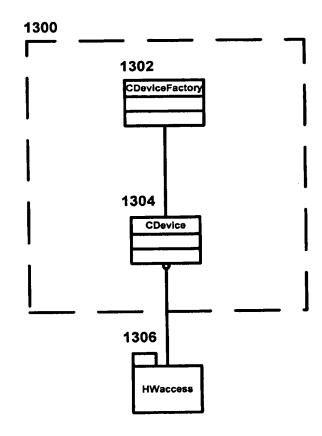


FIG. 35

OBLON, SPIVAK, et al. Docket No: 245418US-2 Inventor: Tetsuro MOTOYAMA, et al. SHEET 38 of 39

Protocol Parameter Map 1400

1402

1404

:		
strin	g	vector <sparameter></sparameter>
strin	9	vector <sparameter></sparameter>
strin	9	vector <sparameter></sparameter>
strin	g	vector <sparameter></sparameter>
• •	•	
i		

1406

struct SParameter {
 std::string m_sParName;
 std::string m_sParValue;
 SParameter ();
 ~SParameter ();
 void clear();
};

FIG. 36A

1410

SNMP	<(COMMUNITY, private)>
FTP	<(USERNAME, abc), (PASSWORD, xyz)>

FIG. 36B

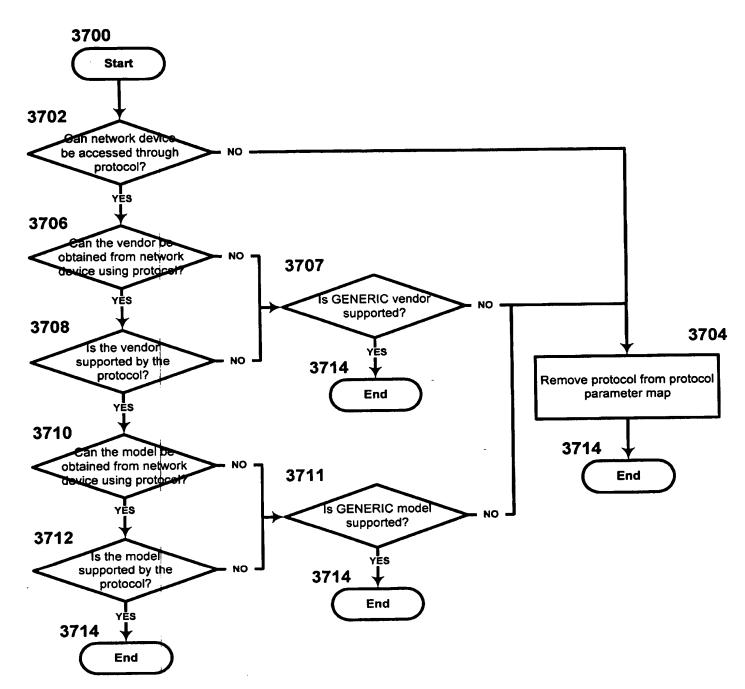


FIG. 37